

*Sebastian Frank, Mathias Weiß, Marion Braunschweig*

## ***XML-basierter Befehlssatz für NPM-Maschinen***

# XML-basierter Befehlssatz für NPM-Maschinen

Sebastian Frank, Mathias Weiß, Marion Braunschweig

## 1 Einleitung

Die zunehmende Miniaturisierung sowohl elektronischer als auch mechanischer Systeme erfordert Werkzeuge, die den Zugang zum Nanometer-Bereich eröffnen. Nanopositionier- und Nanomessmaschinen (NPM-Maschinen) sind Maschinen, die Messungen und Manipulationen an Objekten dieser Größenordnung ermöglichen.

Durch die besonderen Eigenschaften der NPM-Maschinen können diese für eine Vielzahl unterschiedlichster Einsatzgebiete verwendet werden. Die metrologisch optimierte Struktur des mechanischen Gestells und die Einhaltung des Abbeschen Messprinzips in allen drei Raumachsen ermöglicht beispielsweise das Vermessen feinsten Strukturen im Nanometerbereich. Durch den großen Messbereich der Maschinen sind derartige Messungen sogar an makroskopischen Objekten an unterschiedlichen Positionen und mit einer hinreichend großen Wiederholgenauigkeit möglich. Die Entwicklung neuer Bearbeitungstools, so genannter Nano-Tools, erweitern die Einsatzgebiete von NPM-Maschinen nochmals. Die Integration einer „optischen Pinzette“ beispielsweise, ermöglicht die Manipulation nanoskaliger Objekte im Raum wie dies unter anderem für biologische Anwendungsfälle notwendig ist.

Entsprechend der großen Zahl unterschiedlicher Einsatzgebiete von NPM-Maschinen existieren auch unterschiedlichste Anforderungen ihrer Nutzer hinsichtlich der Bedienung. Je nach Einsatzzweck werden ganz andere Bedingungen gestellt oder ganz andere Funktionen verlangt. Die Steuerung von NPM-Maschinen muss daher flexibel bzw. modular gestaltet sein, um den jeweiligen Bedürfnissen angepasst werden zu können.

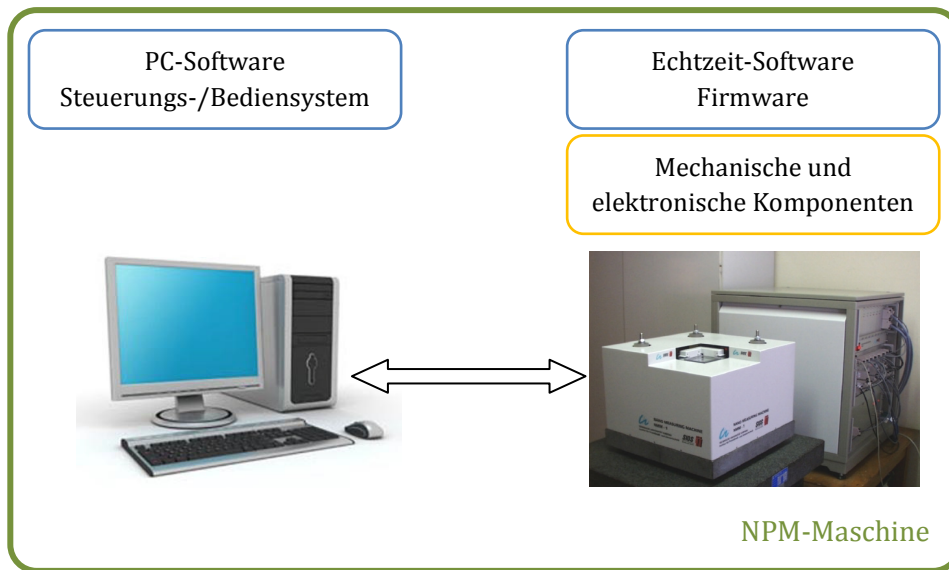
Der vorliegende Artikel soll den Stand der Arbeiten an einem modularen Steuerungskonzept für NPM-Maschinen aufzeigen.

## 2 Aufbau von Nanopositionier- und Nanomessmaschinen

Nanopositionier- und Nanomessmaschinen sind Systeme die sowohl aus mechanischen und elektronischen als auch aus Software-Komponenten bestehen. Bild 1 zeigt die allgemeine Struktur eines solchen Systems. Anschließend wird in Bild 2 das mechanische System von NPM-Maschinen genauer dargestellt.

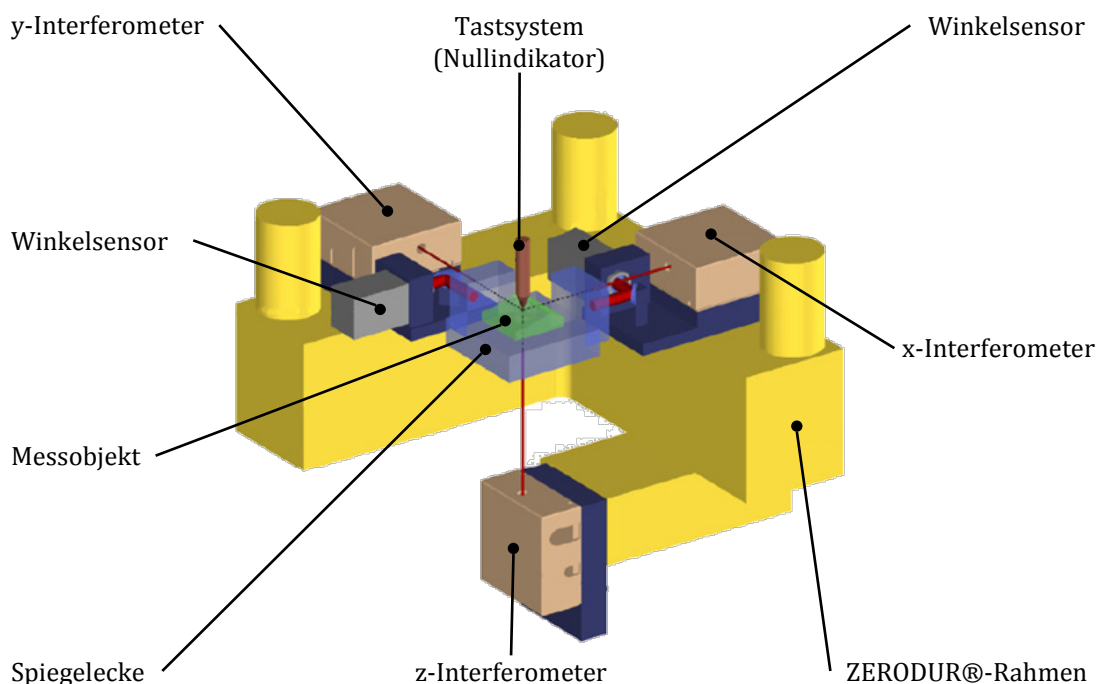
Die eigentliche NPM-Maschine besteht aus dem mechanischen Aufbau mit der zentralen Spiegelecke, den Mess- und Antriebssystemen, einer elektronischen Steuereinheit und der eingebetteten Echtzeitsoftware (Firmware). Das zentrale Element des mechanischen Aufbaues ist ein thermisch stabiler metrologischer Rahmen, das Gestell. Dieses ist aus ZERODUR® gefertigt und bildet die mechanische Basis für die Antriebssysteme. Die Antriebe, realisiert durch elektromagnetische Linearantriebe, ermöglichen eine Bewegung des an ihnen befestigten Messtisches entlang aller drei Raumachsen. Um die Position des Messtisches über den gesamten Messbereich hinweg bestimmen zu können, ist an ihm eine hochpräzise Referenz, die sogenannte Spiegelecke angebracht. Deren Lage im Raum (X-, Y-, und Z-Koordinate, sowie Verkipfung/Verdrehung um die Achsen) wird mit

Hilfe von Laserinterferometern gemessen. Die Lageregelung der Antriebe erfolgt durch die elektronische Steuereinheit.



**Bild 1:** NPM-Maschinen: Ein System aus mechanischen und Software-Komponenten

Bereits aus Bild 1 ist ersichtlich, dass zum Gesamtsystem NPM-Maschine mehrere Software-Komponenten gehören. Zunächst ist hier die Echtzeit-Software der Steuerung zu nennen. Diese verarbeitet direkt die Sensorsignale und übernimmt die Regelung aller dynamischen Prozesse innerhalb der Maschine. Daneben befindet sich eine weitere Gruppe von Software-Komponenten auf dem Bedien-PC. Dieser ist über eine standardisierte Schnittstelle (derzeit USB 2.0) mit der Steuerung verbunden. Der Bedien-PC hat die Aufgabe die Maschine zu steuern, Messaufgaben zu ermöglichen und Messergebnisse auszuwerten bzw. zu verwalten. Im folgenden Kapitel 3 werden die Komponenten auf Seite des Bedien-PCs näher erläutert.



**Bild 2:** Mechanische Komponenten der NPM-Maschine

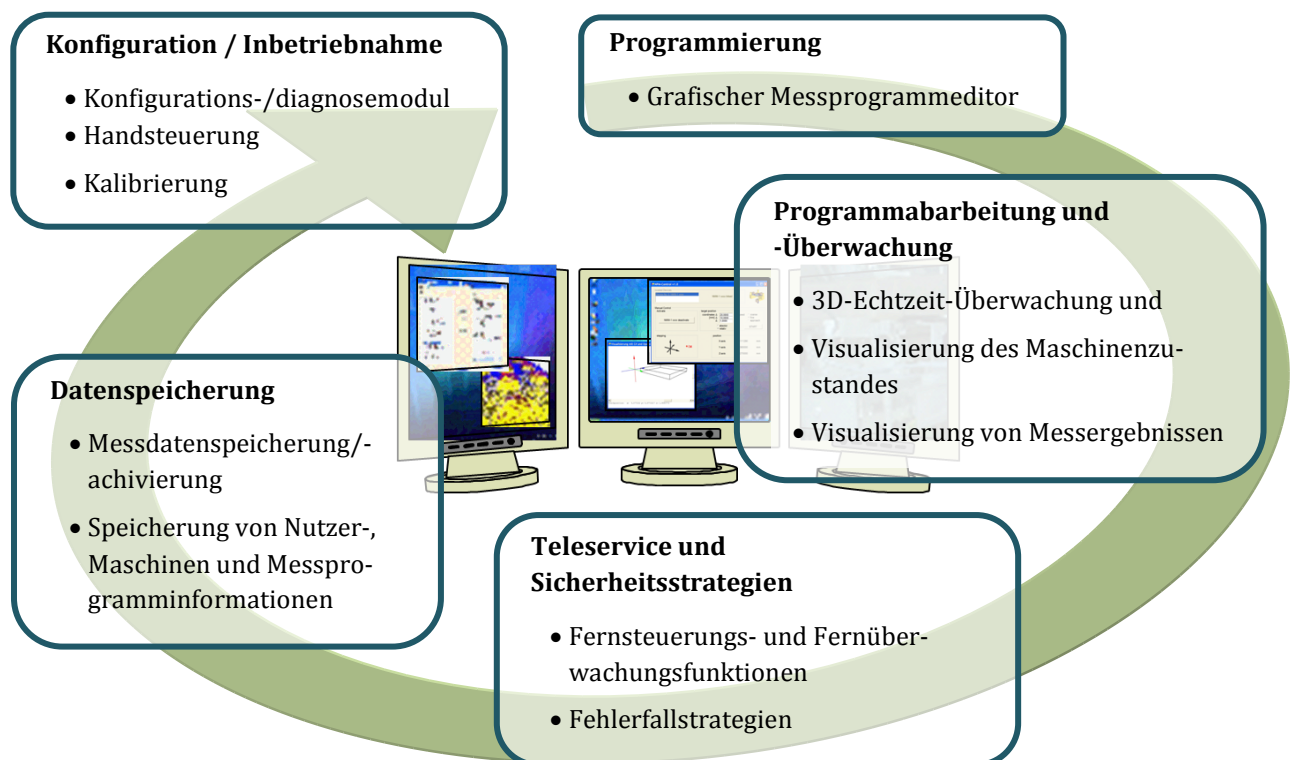
### 3 Softwaremodule von NPM-Maschinen

#### 3.1 Aufgabe der Bediensoftware

Die Arbeitsphasen während der Nutzung von NPM-Maschinen lassen sich anhand folgender Gliederung einteilen:

- Auslieferungs-/Inbetriebnahmephase
- Konfigurationsphase
- Kalibrierung
- Erstellung bzw. Bearbeitung von Messprogrammen
- Echtzeit-/Online-Durchführung und -überwachung von Messvorgängen
- Visualisierung von Messergebnissen
- Speicherung und Archivierung von Messdaten
- Teleservice
- Behandlung von Fehlerfällen

Die Schwerpunkte der Liste verschieben sich je nach Einsatzort, -zweck und Bediener-Gruppe. Je nach Anwendungsfall liegt der Fokus beispielsweise auf der Auswertung der Messergebnisse oder in der Erstellung umfangreicher Messprogramme.

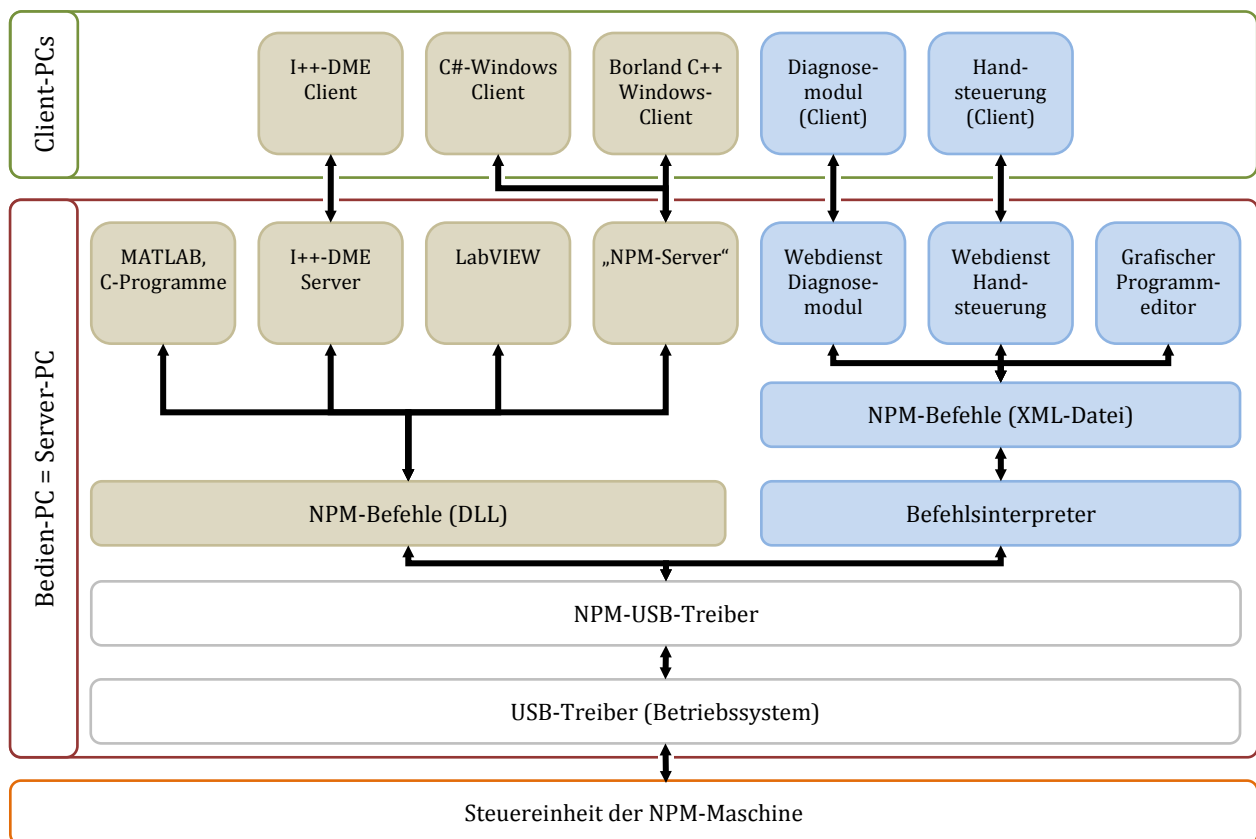


**Bild 3:** Arbeitsphasen und Programm-Module

Um NPM-Maschinen produktiv einsetzen zu können, muss sich die Bediensoftware intuitiv bedienen bzw. an die jeweilige Situation anpassen lassen. Dazu ist es sinnvoll ein modulares Konzept zu verfolgen. Aus Bild 3 geht hervor, welche Module den einzelnen Arbeitsphasen zugeordnet werden können. Im Folgenden werden einige bereits realisierte Module der Bediensoftware vorgestellt.

### 3.2 Softwarestruktur

Neben der eigentlichen Bediensoftware benötigt das Gesamtsystem NPM-Maschine wie bereits erwähnt zusätzliche Software-Komponenten. In Bild 4 ist die gesamte Struktur der Software-Module aufgezeigt. Anhand dieser Abbildung sieht man, dass die Software-Komponenten auf drei unterschiedliche Hardware-Komponenten verteilt sind. Auf den Controllern der Steuereinheit befinden sich spezielle Echtzeitkomponenten. Diese stellen die Firmware der Maschine dar. Die Aufgabe der Echtzeitkomponenten sind vor Allem die Regelung der Antriebe und Messsysteme und die Realisierung einer Schnittstelle für den Zugriff auf die Maschinenfunktionen von außen. Innerhalb dieses Aufsatzes werden die Echtzeitmodule aber nicht näher beschrieben.



**Bild 4:** Softwarestruktur

Die zweite Gruppe der Software-Komponenten befindet sich auf dem Bedien-PC (Operator-PC). Dieser stellt die Nutzerschnittstelle zur Maschine dar. Die Kommunikation aller Software-Komponenten mit der Maschine erfolgt über den betriebssystemintegrierten USB-Treiber. Dieser wird um einen speziellen NPM-USB-Treiber erweitert.

Die Bedien-PC-Software lässt sich in zwei Gruppen unterteilen: Module die auf einer speziellen DLL basieren (grau-braune Kästchen) und Module die auf einem in XML abgebildeten Maschinenbefehlssatz basieren (blaue Kästchen). Die Module, die auf der speziellen NPM-Befehls-DLL basieren, arbeiten mittels Befehlsaufrufen der DLL. Dieses Vorgehen hat zwei entscheidende Nachteile: Er-

weitert sich der Funktionsumfang der Maschinenfirmware, muss eine neue DLL kompiliert und installiert werden. Versionskonflikte sind daher zu erwarten. Ein weiterer Nachteil ist darin zu sehen, dass die DLL nicht hard- und softwareunabhängig ist. Die DLL ist immer an eine bestimmte Rechnerarchitektur und an ein bestimmtes Betriebssystem gebunden. Programme die darauf zugreifen wollen unterliegen somit auch diesen Restriktionen.

Eine andere Variante ist die Verwendung eines auf XML basierenden Befehlssatzes. Dieser bildet alle Befehle in einer bestimmten XML-Struktur ab. So ist es möglich neue Maschinenfunktionen zu ergänzen. Dazu ist lediglich die Erweiterung der XML-Datei nötig. Natürlich muss der Befehl von der Firmware unterstützt werden. Da die XML-Datei nicht installiert werden muss, kann sie immer aktuell und zentral (z.B. über einen bestimmten URL) zur Verfügung gestellt werden. Desweiteren ist diese Variante plattformunabhängig, denn XML-Dateien sind auf jeder Rechner-Architektur nutzbar. Nachteilig an dieser Variante wirkt sich die Notwendigkeit eines zusätzlichen Befehlsinterpreters aus. Dieser muss die Befehle und Funktionen so interpretieren, dass der USB-Treiber entsprechende Funktionsaufrufe übertragen kann. Nähere Informationen zum Funktionsprotokoll sind in [1] zu finden.

Die Unterteilung in DLL- und XML-basierte Module setzt sich schließlich auch in der Gruppe der Client-PC-Komponenten fort. Hier gelten die gleichen Vor- und Nachteile wie sie auch für die Bedien-PC-Komponenten zutreffend sind. Desweiteren bietet der XML-basierte Ansatz auch die Möglichkeit der Realisierung einer plattformunabhängigen Web-Schnittstelle in Form von XML-Webdiensten für einen weltweiten Zugriff auf den Bedien-PC. Dazu siehe auch Kapitel 3.5.

Die in den folgenden Kapiteln beschriebenen Komponenten beziehen sich ausschließlich auf den Ansatz eines XML-basierten Maschinenbefehlssatzes.

### 3.3 XML-basierter NPM-Maschinen-Befehlssatz

Für die Umsetzung des kompletten NPM-Maschinen-Befehlssatzes wurde zunächst der bestehende Befehlssatz auf Basis der „NPM-DLL“ analysiert. Ergebnis ist eine Systematisierung von 58 Befehlen in 9 Befehlsgruppen.

```
<INSTRUCTION name="MOVE_TO_POINT" group_id="17" inst_id="2" txt="Bewegung zu Punkt auf Linie (Nutzer)">
  <PARAMETERS>
    <PARAM name="KoMod" type="tMoveMode" txt="Art der Koordinateneingabe" />
    <PARAM name="ZielKo" type="tDouble3d" txt="Zielkoordinaten (3D-Punkt)" />
    <PARAM name="Geschw" type="tMoveSpeed" txt="Verfahrgeschwindigkeit" />
  </PARAMETERS>
  <RETURN_VALUES>
    <RET_VAL name="FNr" type="Byte" txt="Fehlernummer" />
  </RETURN_VALUES>
</INSTRUCTION>
```

**Bild 5:** Ausschnitt aus dem XML-Befehlssatz

Da die XML-Datei in Zusammenhang mit dem Befehlsinterpreter als eine Art „Interface“ zwischen den höheren Software-Modulen und dem Maschinen-USB-Treiber agiert, ist es für die höheren Softwareschichten im Prinzip nur wichtig, die „Signaturen“ der Maschinenbefehle zu kennen. Wie in Bild 5 zu sehen ist, wird von jedem Befehl genau diese Signatur abgebildet. Die darauf basierenden Programm-Module erhalten dadurch alle notwendigen Informationen über Datentypen der Parameter und eventueller Rückgabewerte. In einem hier nicht abgebildeten Teil des Befehlssatzes sind weitere Informationen, wie beispielsweise die Art des Aufrufes (synchron oder asynchron) hinterlegt, die dem Interpreter Möglichkeiten einer Ablaufsteuerung geben.

Wie bereits erwähnt, ist es durch diese Vorgehensweise möglich neue Maschinenbefehle in der Datei zu implementieren. Dies kann theoretisch mit jedem beliebigen Text-Editor erfolgen. Um die Konsistenz der Datei zu erhalten, wird zusätzlich zur XML-Befehlsdatei eine XML-Schemadatei bereitgestellt, mit deren Hilfe eine Überprüfung der Befehlsdatei erfolgen kann. Diese Technik ist durch das W3C standardisiert.

Der größte Vorteil bei der Verwendung des XML-basierten Befehlssatzes zeigt sich in der Plattformunabhängigkeit. Jedes Programmiermodell welches in der Lage ist XML-Datei zu lesen und zu erstellen, kann nun mit der NPM-Maschine kommunizieren. Die Verwendung des XML-basierten Befehlssatzes ermöglicht es also, eine Vielzahl verschiedener Programm-Module für die Arbeit mit NPM-Maschinen für die unterschiedlichsten Einsatzfälle zu erstellen.

### 3.4 Grafischer Messprogrammeditor

Eines der Hauptanwendungsgebiete der NPM-Maschine ist das Vermessen von Strukturen im Nanometerbereich. Dazu ist Messprogramm notwendig, welches den gesamten Messvorgang steuert. Bei der Verwendung eines taktilen Messtasters besteht ein Messvorgang beispielsweise mindestens aus den Teilprozessen:

- Annähern an die Objektoberfläche
- Objekt antasten
- Messwert aufnehmen
- evtl. Aufnahme weiterer Messpunkte
- Messtaster von Oberfläche abheben
- ...

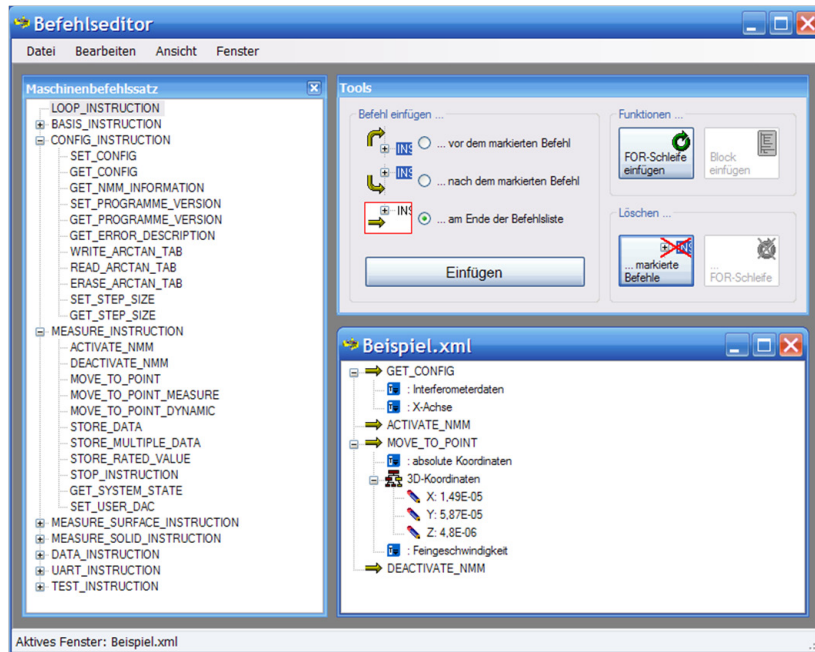
Wird die Messung gestartet, werden die Teilprozesse automatisch als Messprogramm abgearbeitet. Der grafische Messprogrammeditor stellt daher ein zentrales Modul des modularen Bediensystems dar.

Ziel des Messprogrammeditors ist es, derartige Messprogramme zu erstellen oder zu editieren. Da die verfügbaren Maschinenbefehle übersichtlich in einer Baumstruktur dargestellt werden, ist die Einarbeitung in die Programmierung der NPM-Maschine schnell und ohne Kenntnis einer Programmiersprache möglich. Zusätzlich kann durch geeignete XML-Transformationen eine spezifische Befehlsliste generiert werden, die durch Verbergen einiger Funktionen speziell an den Nutzerkreis oder die Arbeitsphase angepasst sein kann. Aus der Liste der Maschinenbefehle wählt der Nutzer entsprechende Funktionen aus und setzt sie zu einem kompletten Messprogramm zusammen. Notwendige Parameter-Einstellungen werden dialogbasiert abgefragt und können auch später noch verändert werden. Bild 6 zeigt ein Beispiel für ein Messprogramm welches mit dem Editor erstellt wurde.

Neben den eigentlichen Maschinenbefehlen bietet der Messprogrammeditor auch Funktionen zur Programmablaufsteuerung. Dazu gehören Verzweigungen, Schleifen und die Möglichkeit existierende Programmblöcke einzufügen. Erst mit Hilfe dieser Ablauffunktionen wird die Erstellung komplexer Messprogramme oder Scanvorgänge möglich.



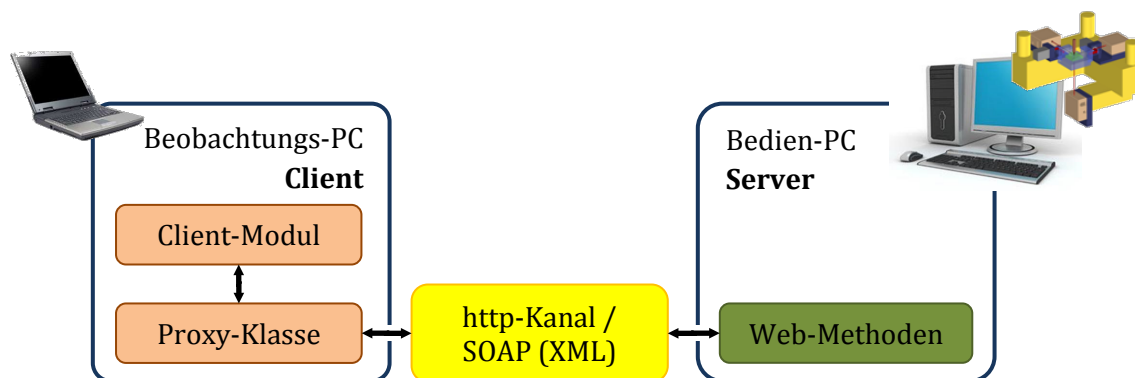
Im Ergebnis der Programmerstellung erzeugt der Messprogrammeditor wiederum eine XML-Datei. Diese wird vor jeder Weiterverarbeitung stets gegen ein weiteres XML-Schema geprüft, um die Konsistenz des Programms sicherzustellen. Nach dem Erstellen des Programms wird dieses über den Interpreter an die Maschine gesendet und die resultierenden Messergebnisse werden aufgezeichnet.



**Bild 6:** Screenshot der Bedienoberfläche des grafischen Messprogrammeditors

### 3.5 Teleservice-Funktionen

Oft ist es erforderlich, dass die Bedienung der Maschine auch von einem entfernten Ort aus erfolgen kann. Dazu ist der Zugriff auf die Maschinenfunktionen über das Internet hinweg nötig. In den letzten Jahren haben sich dafür zahlreiche Standardtechnologien etabliert. Eine davon sind die sogenannten XML-Webdienste. Wie der Name dieser Technologie bereits vermuten lässt, basiert der Datenaustausch bei diesem Verfahren auf dem Transport von XML-kodierten Informationen. Das dabei verwendete Protokoll heißt SOAP (Simple Object Access Protocol). Eine Prinzipskizze ist in Bild 7 abgebildet.



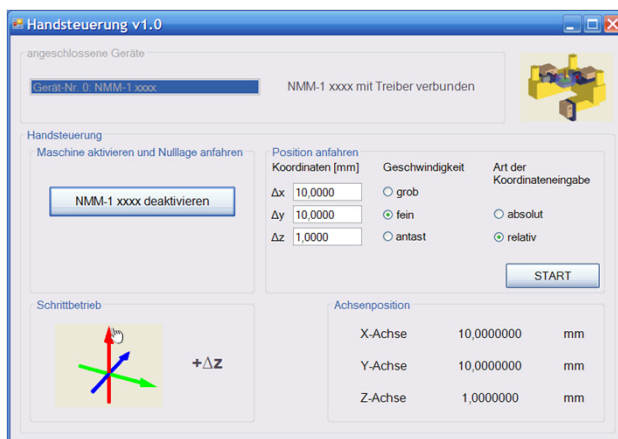
**Bild 7:** Prinzip von XML-Webdiensten

Vereinfacht ausgedrückt kann ein Client-Programm mit Hilfe dieser Technologie Funktionsaufrufe so durchführen, als ob die Ausführungsdomäne dieser Funktionen auf dem eigenen Rechner liegen



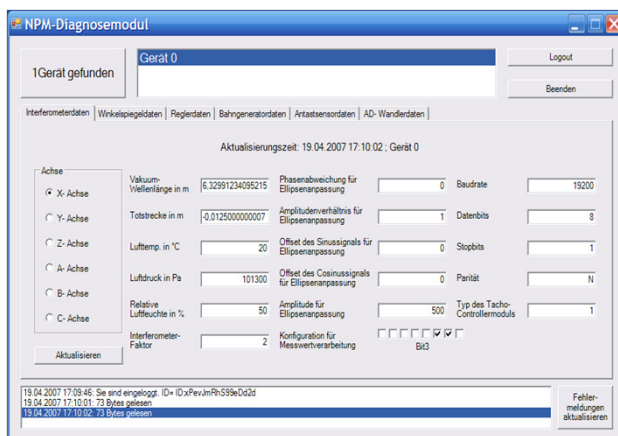
würde. Der Datenaustausch erfolgt dabei vollkommen transparent über einen http-Kanal, der die XML-Pakete überträgt.

Für einen Demonstrator des modularen Bediensystems für NPM-Maschinen wurden bereits zwei Clients realisiert, die mittels XML-Webdiensten auf die Maschine zugreifen. Zum einen ist das eine Handsteuerung. Diese erlaubt es, die Antriebe der Maschine zu steuern und damit den Messtisch zu bewegen. Dies kann entweder schrittweise oder durch Vorgabe einer Zielposition erfolgen. Siehe dazu auch Bild 8. Der Client „Handsteuerung“ zeigt einen weiteren Vorteil der Technologie Webdienste: Es ist nämlich möglich, den Client auch auf dem Bedien-PC laufen zu lassen. Für die Programmierung der Programm-Module ist es daher unerheblich ob sie für den Client oder den Bedien-PC geschrieben werden. Es muss lediglich ein Programm geschrieben werden (der Client). Dieses kann dann aber sowohl auf dem Server als auch dem Client ausgeführt werden.



**Bild 8:** Handsteuerung

Der zweite bereits realisierte Client ist ein Diagnosemodul. Mit dessen Hilfe lassen sich jederzeit von jedem beliebigen Ort Maschinenparameter auslesen und gegebenenfalls auch ändern. Die Programmoberfläche dieses Clients zeigt Bild 9.



**Bild 9:** Diagnose-Modul

## 4 Zusammenfassung und Ausblick

Aufgrund der zahlreichen Anwendungsgebiete von NPM-Maschinen werden sehr komplexe Anforderungen an das Bediensystem dieser Maschinen gestellt. Ein solches System muss die Anforder-

rungen unterschiedlichster Anwendergruppen berücksichtigen. Dies ist nur durch ein modular anpassbares Bediensystem zu erreichen. Der Artikel zeigt erste Module für eine PC-basierte Steuerung einer NPM-Maschine auf Basis eines in XML abgebildeten Maschinenbefehlssatzes.

Als Basis für den Datenaustausch zwischen den Modulen hat sich XML als vielversprechend erwiesen und wird daher konsequent genutzt. Das zeigt sich in der Verwendung dieses Datenformates als Basis für den Maschinenbefehlssatz, als Speicherformat für Messprogramme und in der Verwendung der Remote-Technologie XML-Webdienste.

Weitergehende Arbeiten müssen sich mit der Integration dieser Module in ein Framework beschäftigen, um das Ziel einer modularen Programmstruktur zu erreichen.

## 5 Quellenverzeichnis

- [1] **Frank, Sebastian:** Entwurf einer modularen Steuerung für Nanopositionier- und Nanomessmaschinen. Dissertation. Universitätsverlag Ilmenau. Ilmenau. 2007
- [2] **Frank, S.; Weiß, M.; Braunschweig, M.; Birli, O.:** PC-software modules for a NPM machine control system. Portrait - Faculty of Mechanical Engineering: proceedings - 53rd IWK, ICPM 2008. Ilmenau. 2008
- [3] **Weiß, M.; Braunschweig, M.; Birli, O.; Frank, S.:** Modulares Operatorsystem für Nanopositionier- und Nanomessmaschinen. Tagungsband ‚Mikrosystemtechnik Chemnitz 2009‘. Chemnitz. 2009